

Integrando GGL en servidores WPS transaccionales

V. González ⁽¹⁾, B. Schäffer ⁽²⁾ y F. González ⁽³⁾

⁽¹⁾ Universidad Politécnica de Valencia, Camino de Vera, s/n, 46022 Valencia, vicgonco@upv.es

⁽²⁾ Geoprocessing Community, 52° North Initiative for Geospatial Open Source Software GmbH, Martin-Luther-King-Weg, 24, 48155 Münster, schaeffer@uni-muenster.de

⁽³⁾ Ingeniero Informático, Autónomo, fernando@fergonco.es

RESUMEN

Este texto presenta los principales resultados de la integración del lenguaje de geoprocésamiento proporcionado por GearScape: GGL (Gearscape Geoprocessing Language) en la extensión transaccional del estándar WPS implementada por la iniciativa 52° North (WPS-T).

Palabras clave: WPS, transaccional, estándar, SIG, lenguaje de geoprocésamiento, script, SQL espacial.

ABSTRACT

This paper presents the main results of the integration of the geoprocessing language provided by Gearscape: GGL (Gearscape Geoprocessing Language) into the transactional extension of the WPS standard implemented by 52° North (WPS-T).

Key words: WPS, transactional, standard, GIS, geoprocessing language, script, spatial SQL.

INTRODUCCIÓN

Al principio de los 90, (Franklin, 1992) se concluyó que “sobre el ochenta por cien de los datos almacenados en bases de datos corporativas tenían una componente espacial” [1]. Desde entonces, ha habido multitud de inversiones en el desarrollo de infraestructuras de datos espaciales [2][3] aunque la mayor parte de ellas se han centrado en la obtención de datos y la visualización [4]. Estas inversiones, además de las técnicas de adquisición de datos mejoradas, han incrementado la cantidad y el tamaño de los conjuntos de datos disponibles y, por tanto, la complejidad de los nuevos modelos y los análisis hechos con ellos.

Por otra parte, los lenguajes de procesamiento son herramientas útiles para definir y comunicar análisis sobre datos espaciales. En particular, el lenguaje de consulta estructurado (SQL) extendido con la norma OGC para el acceso a fenómenos simples

es uno de los más adecuados para la realización de análisis espacial [5][6]. Principalmente, su capacidad de especificar geoprocursos en *scripts* puede ser muy útil para documentar, compartir y desarrollar colaborativamente análisis espaciales.

En el presente artículo se mostrará la integración del lenguaje de geoprocusamiento de GearScape en un servidor WPS para así añadir las características que tienen los lenguajes de geoprocusamiento a las infraestructuras de datos espaciales.

CONTEXTO

La especificación del estándar OGC Web Processing Service (WPS) [7] detalla un conjunto de operaciones para la publicación y la ejecución de cualquier tipo de geoprocuro en la web. Según la definición de este estándar, un proceso se define como un algoritmo, cálculo o modelo que opera con datos geográficamente referenciados.

No obstante, el estándar WPS proporciona un esqueleto vacío en el que se deben incluir implementaciones concretas de los algoritmos. Esto, unido a que la interfaz WPS no proporciona ningún medio para desplegar procesos en tiempo de ejecución ha llevado a la propuesta de una extensión transaccional (WPS-T) [8]. Esta extensión permite la definición de perfiles de despliegue que determinan la forma en la que un tipo de proceso específico debe de ser codificado para ser entendido y desplegado mediante la interfaz WPS-T. De esta manera, procesos que anteriormente sólo eran accesibles en un entorno específico y tenían muy poca o ninguna visibilidad fuera de dicho entorno están al alcance de cualquiera que tenga acceso a Internet, lo cual mejora considerablemente la accesibilidad. La promesa de compartir modelos geoespaciales [9] se convierte así en realidad de una forma estandarizada para poder ser incluida en las infraestructuras de datos espaciales.

Por otra parte, GearScape es un Sistema de Información Geográfica de escritorio que proporciona un lenguaje para definir geoprocursos: GGL (Gearscape Geoprocessing Language). Este lenguaje implementa un subconjunto de las características del estándar SQL92 [10] y se extiende espacialmente mediante la especificación OGC para fenómenos simples [11][12].

Entre las principales ventajas de este lenguaje encontramos, por un lado, la simplicidad que proporciona un lenguaje declarativo como SQL que, a su vez, disminuye en gran medida la posibilidad de incluir errores en la creación de procesos [13]. Por otra parte, cabe destacar que los motores SQL en general, y GGL en particular determinan automáticamente una estrategia optimizada de procesamiento. Además, GGL extiende el estándar SQL para posibilitar la definición de parámetros de entrada [13]; de esta manera, los geoprocursos pueden ser creados y encapsulados por usuarios con conocimientos del lenguaje y reutilizados por usuarios sin dicho conocimiento. Fue esto precisamente lo que permitió la inclusión de los *scripts* de geoprocusamiento en los servidores WPS transaccionales. Por último, cabe destacar también que con el fin de facilitar la encapsulación y reutilización de *scripts*, GearScape proporciona un editor con validación sintáctica y semántica, asistentes para la ejecución y un modelador para componer geoprocursos en cadenas más complejas.

INTEGRACIÓN

La integración del lenguaje de geoprocusamiento GGL en los servidores WPS transaccionales ha pasado esencialmente por la definición de un perfil de despliegue concreto para este lenguaje, tal y como se detalla en [8]. Este perfil consiste en definir

un esquema XML que contiene un único elemento con el *script* de geoprocetamiento en texto plano; puesto que el *script* es autodescriptivo, este esquema es suficiente para desplegar el proceso en el servidor y, posteriormente, ejecutarlo correctamente, tal y como se explica en [13].

Además, junto con esta integración se han hecho otros desarrollos complementarios que consisten en proporcionar un mayor abanico de posibilidades en el lado del cliente. Así, se ha desarrollado un cliente completamente nuevo en GearScape para desplegar y ejecutar procesos y el cliente de ejecución existente en uDig se ha mejorado para poder acceder de manera transparente a los nuevos geoprocetamientos.

RESULTADOS

Como prueba de concepto del trabajo realizado se han creado dos casos prácticos que ponen de manifiesto la reusabilidad e interoperabilidad del sistema. Básicamente, el flujo de trabajo consiste en que, en primer lugar, un usuario con conocimientos del lenguaje crea un *script* de geoprocetamiento. Posteriormente, este usuario con una instancia de la implementación WPS-T de 52° North para incluir el proceso y, finalmente, cualquier otro usuario (o él mismo) ejecuta el proceso desde GearScape, uDig o cualquier otro cliente WPS. Siguiendo este flujo de trabajo, en los siguientes subapartados se describen de manera detallada ambos casos de uso.

Reusabilidad

Para este caso práctico imaginaremos que somos responsables de Obras Públicas en una comunidad determinada en la que se van a construir ciertos tramos de carretera. Disponemos de un fichero con estos tramos de carretera y otro con datos sobre nuestra comunidad. Nuestra tarea consiste en determinar las provincias afectadas por la nuevas obras y tomar las medidas correspondientes. Posteriormente, las provincias afectadas, demostrarán la reusabilidad haciendo uso del mismo geoprocetamiento para obtener información de los municipios afectados. Para esto, hacemos uso de GearScape y utilizamos el siguiente mapa:

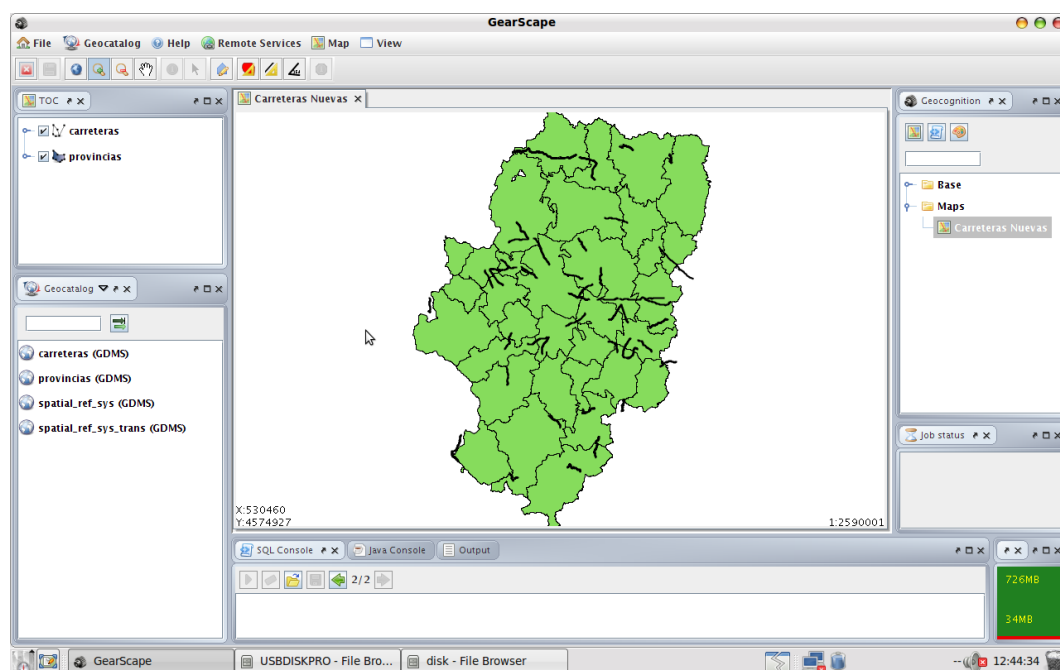


Figura 1: Mapa de nuevas carreteras

Además, supondremos que tenemos algunos conocimientos del lenguaje y hemos creado el siguiente *script* de geoprocésamiento:

```
-- Obtiene informacion de los lugares afectados por las nuevas
carreteras
DOC('Lugares potencialmente afectados por las nuevas carreteras');
DECLARE lugares:TABLE(poligonos:GEOMETRY);

DOC('Nuevas carreteras');
DECLARE carreteras:TABLE(lineas:GEOMETRY);

DOC('Margen de distancia');
DECLARE margen:INT;

-- Fin de la seccion de parametros

CREATE TABLE buffer AS SELECT Buffer(lineas, margen, 'ROUND') AS
the_geom FROM carreteras;
CREATE TABLE "lugaresAfectados" AS SELECT poligonos AS the_geom,
lugares.* FROM lugares, buffer WHERE
Intersects(buffer.the_geom, poligonos);
DROP TABLE buffer;
```

Básicamente el *script* consiste, en primer lugar, en una línea con la descripción general del *script*. Posteriormente, hay una sección en la que se declaran los parámetros de entrada (*DECLARE*), precedidos de su descripción (*DOC*); en nuestro caso se declaran dos tablas como parámetros de entrada (*lugares* y *carreteras*) con sendos campos de tipo geométrico y un *margen* de distancia que incluiremos en el cálculo para obtener las regiones a menos de dicha distancia de las nuevas carreteras.

Posteriormente, tras la sección de parámetros, encontramos la sección de operaciones. En nuestro caso, en primer lugar se crea un *buffer* alrededor de las nuevas carreteras que tiene como tamaño el margen especificado como parámetro. En segundo lugar, se crea la tabla *lugaresAfectados* que tiene un campo *the_geom* consistente en la intersección de los lugares y el *buffer* calculado sobre las carreteras, junto con el resto de información de dichos lugares, especificada en el parámetro de entrada.

Finalmente, se eliminan (*DROP*) las tablas utilizadas como resultados intermedios; en nuestro caso, únicamente necesitamos la tabla *lugaresAfectados* y eliminamos la tabla intermedia *buffer*.

Una vez tenemos el *script* de geoprocésamiento en GGL, podríamos ejecutarlo en nuestra propia máquina. Sin embargo, disponemos de un servidor WPS transaccional con mayor potencia computacional que nuestra máquina y que podría estar haciendo uso de técnicas *grid* para el cálculo en paralelo. Así pues, el siguiente paso consiste en desplegar nuestro *script* en dicho servidor. Haciendo uso del cliente de GearScape, solo necesitamos especificar la URL del servidor y el identificador de nuestro proceso.

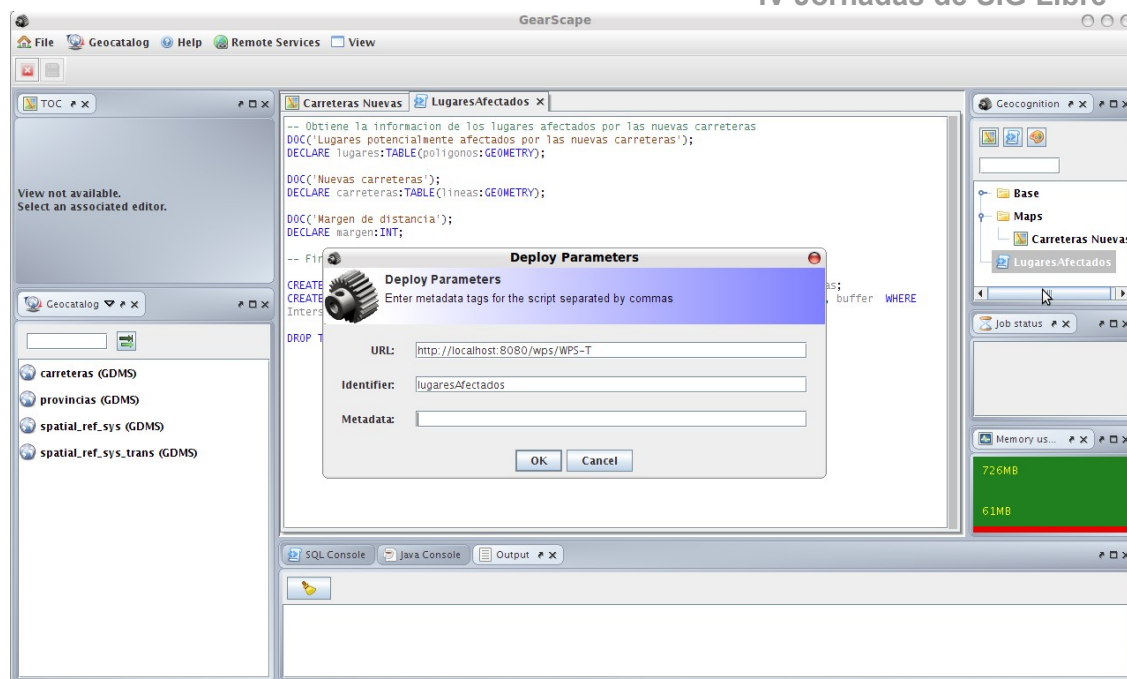


Figura 2: Despliegue de scripts

Una vez nuestro geoproceto está incluido en el servidor, podemos ejecutarlo haciendo uso del nuevo cliente de ejecución de GearScape tal y como si fuera un geoproceto local:

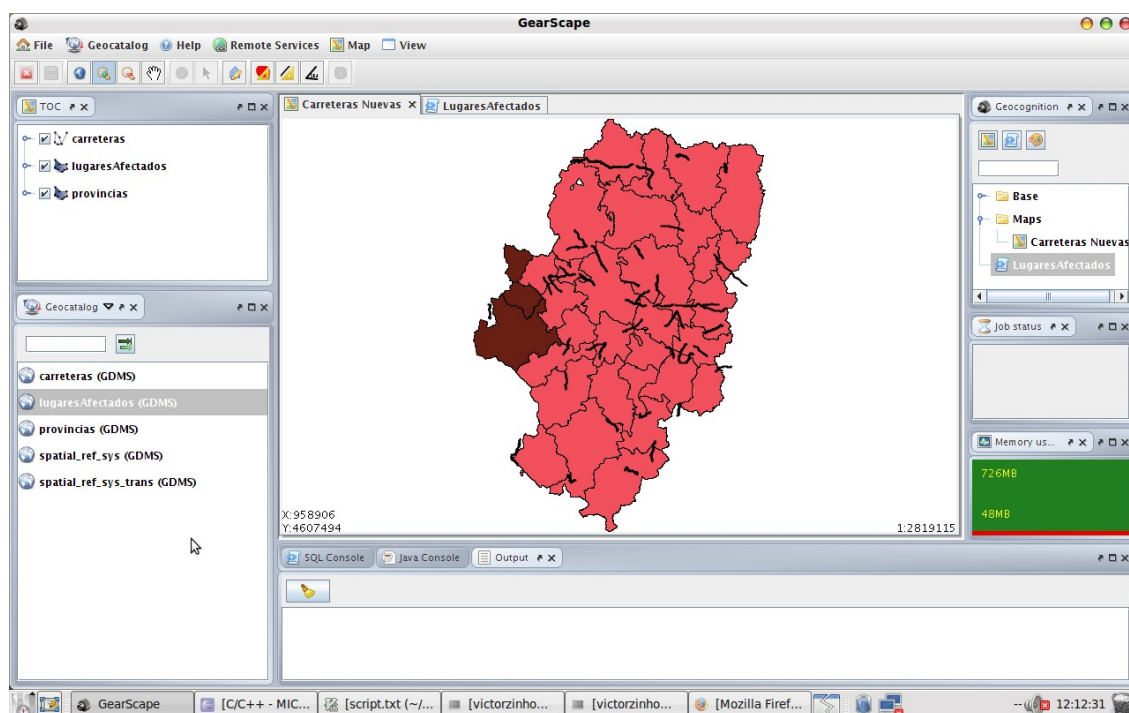


Figura 3: Ejecución del geoproceto sobre provincias

En este punto, y con el fin de mostrar la reusabilidad de los *scripts*, asumiremos que nuestro trabajo como responsable de Obras Públicas consistía simplemente en avisar al responsable en cada provincia, proporcionarle el fichero con las nuevas carreteras y comunicarle la existencia del nuevo geoproceto. Así, el responsable de

la provincia, que no tiene ningún conocimiento del lenguaje, es capaz de crear su propio mapa con los municipios de su provincia y las nuevas carreteras, y ejecutar el nuevo geoproceso de manera transparente para obtener la información necesaria sobre los municipios afectados:

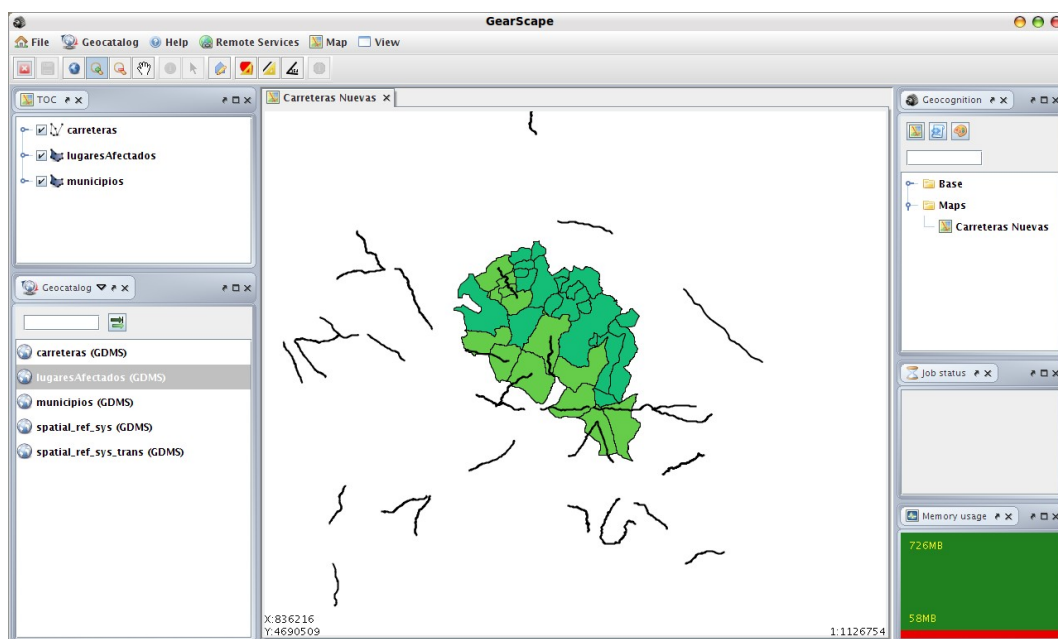


Figura 4: Ejecución del geoproceso sobre municipios

Interoperabilidad

En este caso, haremos uso del siguiente mapa, que contiene un conjunto de núcleos urbanos y servicios o puntos de interés (hospitales, gasolineras y monumentos) en capas diferentes:

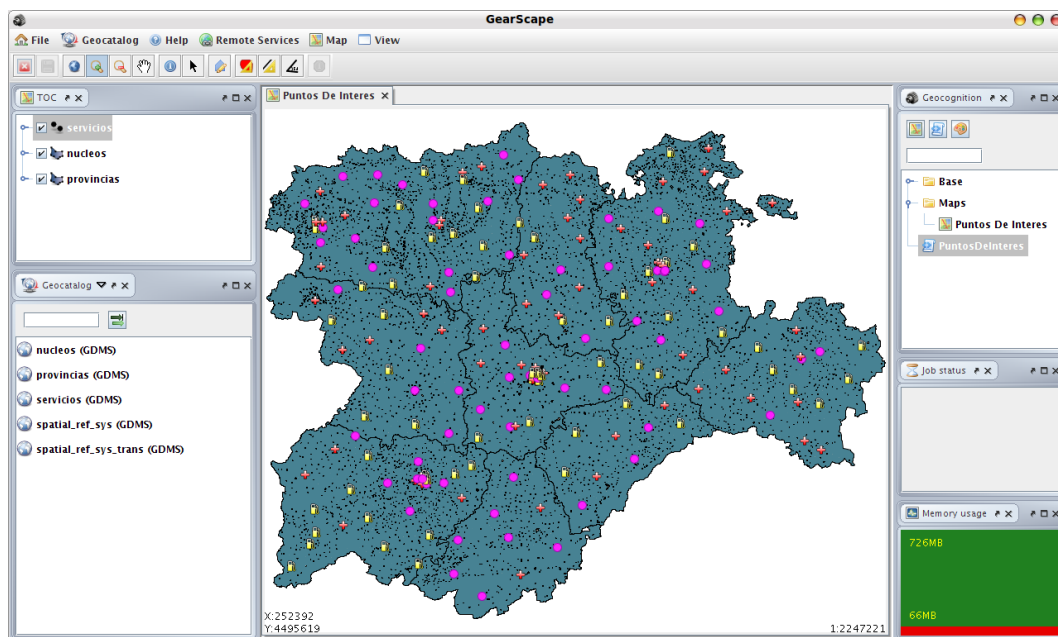


Figura 5: Mapa de servicios

A partir de este mapa obtendremos los núcleos urbanos con hospitales a menos de 10 km. de distancia haciendo uso del cliente de ejecución de GearScape y, posteriormente, de uDig. Para esto, asumiremos que disponemos del siguiente *script* de geoprocésamiento ya incluido en un servidor WPS transaccional:

```
-- Obtiene los nucleos con los servicios especificados a menos de
una distancia determinada
DOC('Nucleos');
DECLARE nucleos:TABLE(nucleo:GEOMETRY);

DOC('Servicios');
DECLARE servicios:TABLE(servicio:GEOMETRY, tipo:ANY);

DOC('Tipo de servicio a obtener');
DECLARE filtro VALUES FROM tipo;

DOC('Distancia maxima');
DECLARE distancia:DOUBLE;

-- Fin de la seccion de parametros
CREATE TABLE count_table AS SELECT nucleos.nucleo, COUNT(nucleo) AS
count FROM nucleos, servicios WHERE tipo = filtro AND
distance(nucleo, servicio) < distancia GROUP BY nucleo;
CREATE TABLE resultado AS SELECT n.*, c.count FROM nucleos n,
count_table c WHERE n.nucleo = c.nucleo ORDER BY c.count;
DROP TABLE count_table;
```

Como se puede observar, en la sección de parámetros se declaran dos tablas con campos de tipo geometría para los *núcleos* de población y los *servicios*. Además, para la tabla de servicios se determina también un campo de cualquier tipo (*ANY*), que contendrá los valores que se usarán para clasificar los tipos de servicio mediante el parámetro *filtro*. Por último, se declara una *distancia* máxima entre los núcleos de población y los servicios.

Posteriormente, en la sección de operaciones, en primer lugar se crea una tabla que contiene el campo geométrico de los núcleos de población junto con el número de servicios del tipo especificado a menos de la distancia máxima determinada. En segundo lugar se fusiona esta tabla intermedia con la tabla de núcleos completa, ordenándola por el número de servicios. Cabe destacar que en esta fusión, ambas tablas comparten el campo geométrico de los núcleos de población y éste aparece tan solo una vez en el resultado. Finalmente, se eliminan los resultados intermedios.

Tras la ejecución del geoprocésamiento mediante GearScape, el resultado obtenido es el siguiente:

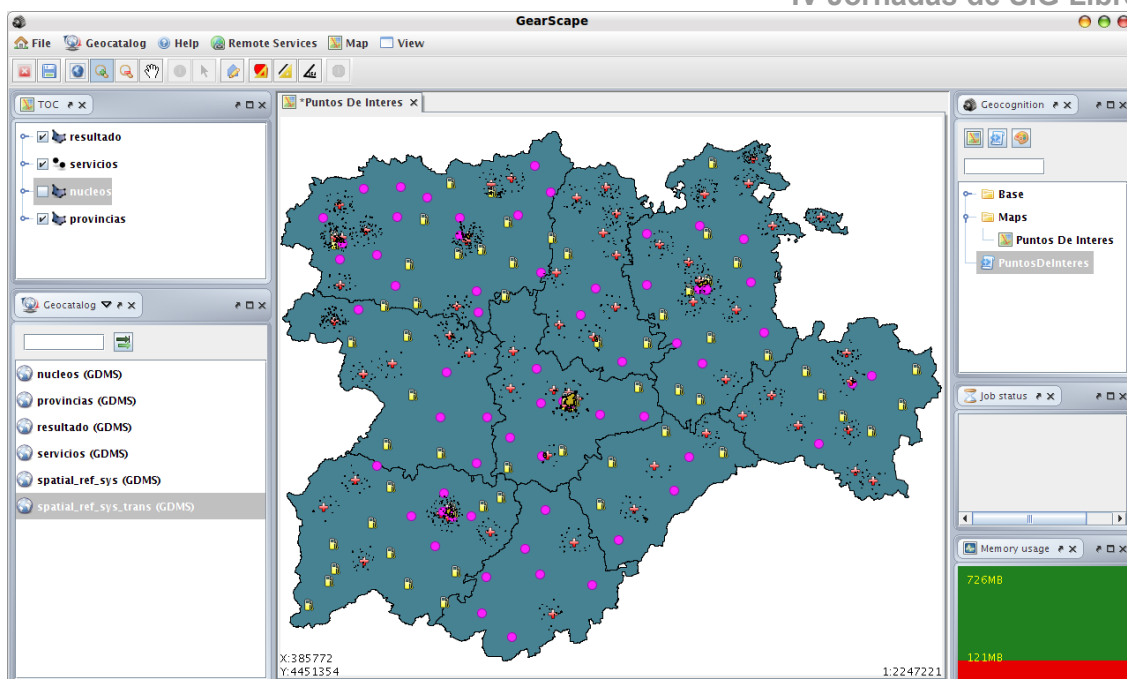


Figura 6: Resultado de la ejecución en GearScape

Y observando más de cerca las cercanías del hospital más meridional:

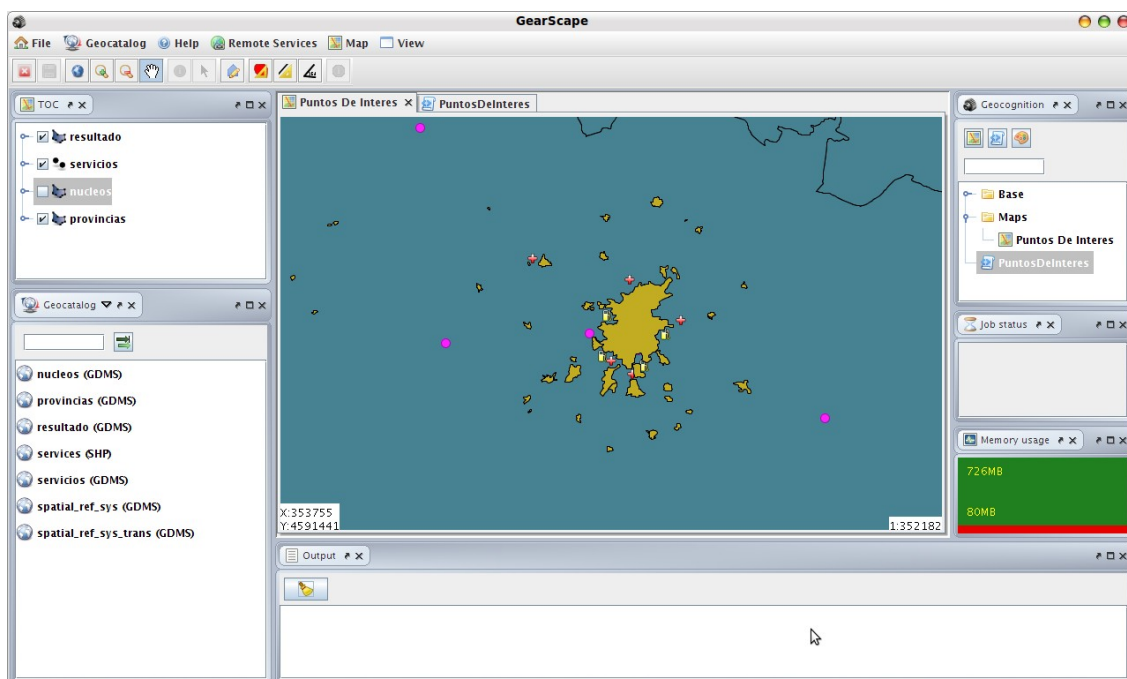


Figura 7: Resultado de la ejecución en GearScape (zoom)

Si ejecutamos el mismo geoproceto con las mismas fuentes de datos en el cliente de ejecución de uDig, el resultado obtenido es, como se esperaba, idéntico. De esta manera, se demuestra la interoperabilidad proporcionada por el sistema que, por otra parte, resulta lógica debido al uso de estándares para el despliegue y la ejecución de geoprocetos.

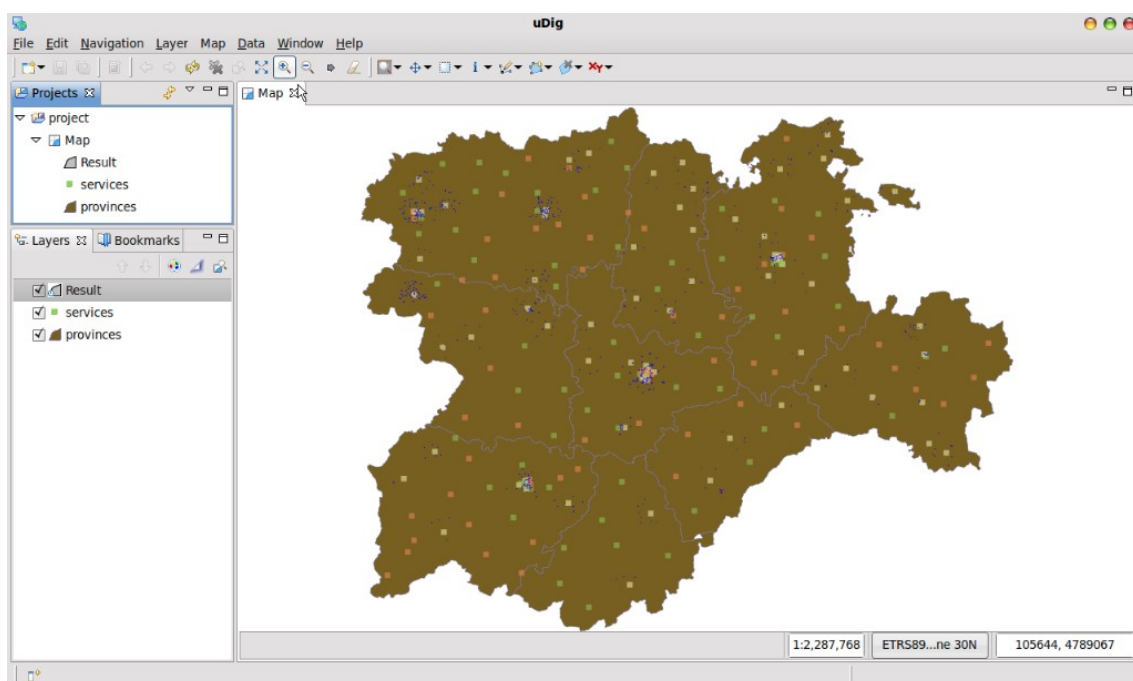


Figura 8: Resultado de la ejecución en uDig

Y acercándonos a las cercanías del mismo hospital que anteriormente:

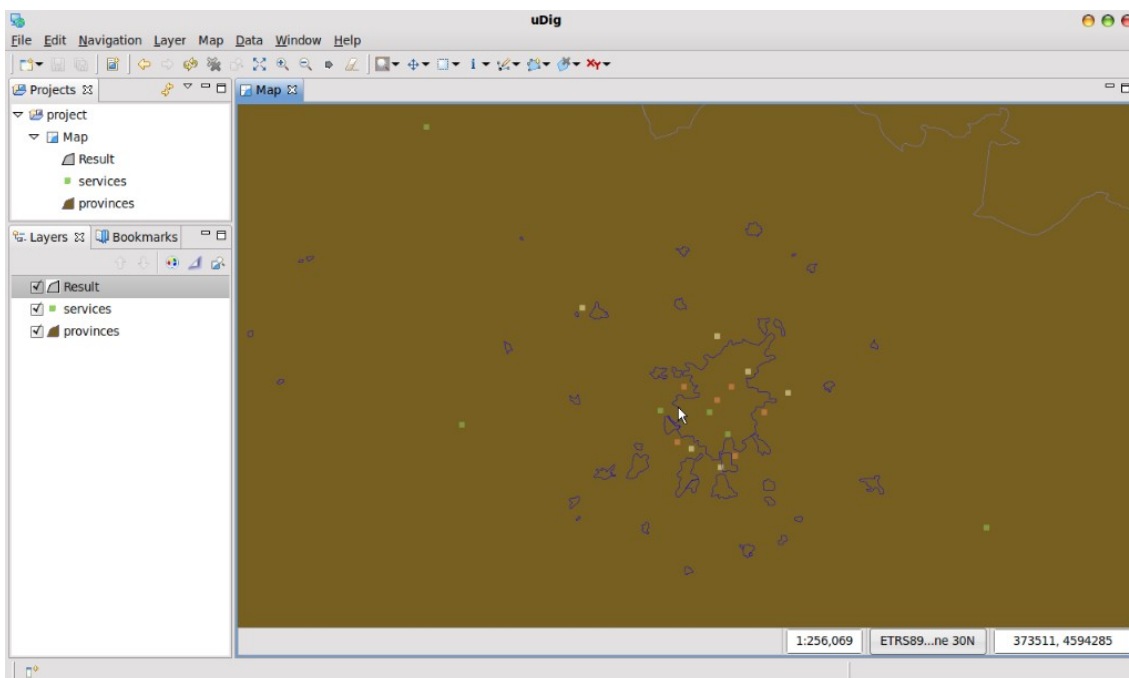


Figura 9: Resultado de la ejecución en uDig (zoom)

TRABAJO FUTURO

Entre los trabajos futuros encontramos, por un lado, convertir GearScape en un Sistema de Gestión de Bases de Datos completo que posibilite la definición de elementos semánticos formados por diferentes estructuras de datos de manera que se puedan definir *scripts* GGL sobre estos elementos y posteriormente incluirlos en los servidores WPS transaccionales. De esta manera sería posible realizar geoprocésamiento con GGL en modelos tales como CityGML.

Por otro lado, también se está empezando a trabajar en la paralelización del motor con el fin de procesar grandes cantidades de datos en entornos *grid* y así reducir tanto los tiempos de ejecución como las limitaciones de memoria.

AGRADECIMIENTOS

Víctor González agradece a todo el equipo de 52° North y a los compañeros del IFGI (Institute For GeoInformatics) de la Universidad de Münster por la oportunidad de realizar el trabajo aquí expuesto, especialmente a Bastian Schäffer por su tutela a lo largo de todo el trabajo, a Christoph Stasch por su impagable labor humana y a Ann Hitchcock por la imprescindible labor logística y organizativa.

REFERENCIAS

- [1] FRANKLIN, C. (1992). An introduction to geographic information systems : linking maps to databases. Database, 15(2) : 12–21.
- [2] CLINTON, B. (1994). Executive order 12906: Coordinating geographic data acquisition and access: The national spatial data infrastructure. www.archives.gov/federal-register/executive-orders/pdf/12906.pdf.
- [3] INSPIRE (2007). Directive of the European Parliament and of the Council establishing an Infrastructure for Spatial Information in the European Community, <http://www.ec-gis.org/inspire/>.
- [4] KIEHLE, C. (2006). Business logic for geoprocessing of distributed data, Computers & Geosciences, vol. 32, 2006, pp. 1746–1757.
- [5] BOCHER, E., LEDUC, T., MOREAU, G., and GONZÁLEZ, F. (2008). GDMS : an abstraction layer to enhance Spatial Data Infrastructures usability. In 11th AGILE International Conference on Geographic Information Science — AGILE'2008, Girona, Spain.
- [6] LEDUC, T., BOCHER, E., GONZÁLEZ, F., and MOREAU, G. (2009). GDMS-R: A mixed SQL to manage raster and vector data. In GIS Ostrava 2009 - Symposium on Seamless Geoinformation Technologies, Ostrava, Czech Republic.
- [7] OGC, *OpenGIS Web Processing Service*, Open Geospatial Consortium, 2007.
- [8] SCHÄFFER, B. (2008), "Towards a transactional Web Processing Service", Proceedings of the GI-Days, Münster.
- [9] GHELOT, S.; VERBREE, E. (2006), "Web-based sharing of a geo-processing chain: combination and dissemination of data and services", Proceedings of the International Symposium on Geospatial databases for Sustainable Development, eds. S. Nayak & S.K. Pathan, ISPRS TC Committee IV Geo- databases and Digital Mapping, Goa, India
- [10] ISO, *Structured Query Language – 92, Standardization*, 1992. International Organization for
- [11] OGC, *OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 1: Common architecture*, Open Geospatial Consortium, 2006.

- [12] OGC, *OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 2: SQL option*, Open Geospatial Consortium, 2006.
- [13] GONZÁLEZ, V.; SCHÄFFER, B; GONZÁLEZ, F. (2009), Publicación y uso de scripts SQL en servidores SQL transaccionales, VI Jornadas Técnicas de la IDE de España, Murcia.